PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
SEMESTER GANJIL TAHUN AJARAN 2021-2022

IF130 Dasar-Dasar Pemrograman

Pertemuan 1 – Algoritma dan Flowchart Struktur Kendali Pemilihan

Yustinus Widya Wiratama, S.Kom., M.Sc., OCA
Anak Agung Ngurah Ananda Kusuma, BE(Hons), MEng, PhD
Putri Sanggabuana Setiawan, S.Kom., M.T.I.

Overview

- Program Development
- Algorithm
- Pseudocode
- Flowchart
- Selection Control Structure

Capaian Pembelajaran Mingguan Mata Kuliah (Sub-CPMK):

1. **Sub-CPMK 01:** Mahasiswa mampu memahami bagaimana proses pemecahan masalah menggunakan Algoritma dalam bentuk diagram alur (flowchart) dan pseudocode. (C2)
2. **Sub-CPMK 02:** Mahasiswa mampu menggambar flowchart dengan struktur kendali pemilihan. (C3)

Program Design

- Progamming can be defined as the development of a solution to an **identified problem**, and the setting up of a related series of **instructions** that, when directed through computer hardware, will produce the **results.**
- Leaping straight into the coding phase **without** first designing a proper solution usually results in a program that contains many **errors.**
- A more experienced programmer will **design** a solution to the program first, desk check this solution, and then **code** the program in a chosen programming language.

Basic Steps in The Development of a Program

1. Define the problem
2. Outline the solution
3. Develop the outline into an algorithm
4. Test the algorithm for correctness
5. Code the algorithm into a specific programming language
6. Run the program on the computer
7. Document and maintain the program

Procedural vs Object Oriented Programming

- Procedural programming is based on a **structured**, **top-down** approach to writing effective programs.
- The approach concentrates on **what** a program has to do and involves identifying and organizing the process in the program solution.
- The program is usually broken down into separate tasks or functions and includes:
  - **Top – down development** : a general solution to the problem is outlined first. This outline is then divided gradually into more detailed steps until finally the most detailed levels have been completed.
  - **Modular design**: is connected directly to top-down development, as the steps or subtasks into which the program solution is divided actually form the future modules of the program.

Procedural vs **Object Oriented Programming**

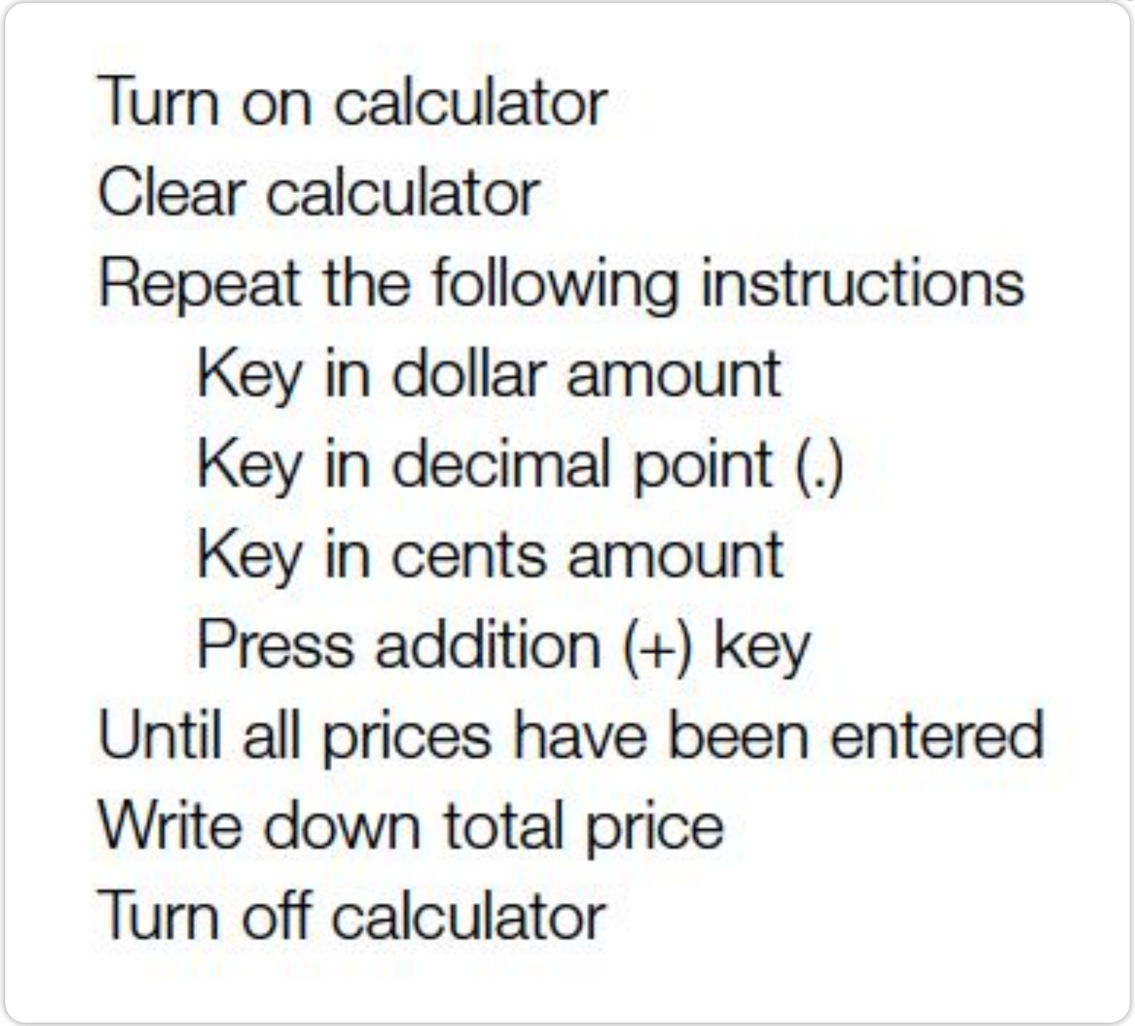
- Object oriented programming is also based on breaking down the problem
- However, the primary focus is on the **things** (or **object**) that make up the program.
- The program is concerned with how the objects behave, so it breaks the problem into a set of separate objects that perform actions and relate to each other.
- These objects have definite properties, and each object is responsible for carrying out a series of related tasks.

What is an algorithm?

- An algorithm is like a **recipe**: it lists the steps involved in accomplishing a task.
- It can be defined in programming terms as a set of *detailed, unambiguous and ordered* instructions developed to describe the **processes** necessary to produce the desired **output** from a given **input**.
- **Flowchart** and **pseudocode** are both popular ways of representing algorithms.

Algorithm Example

- If you want to instruct someone to add up a list of prices on a pocket calculator, you might write an algorithm such as the following:



```
Turn on calculator
Clear calculator
Repeat the following instructions
    Key in dollar amount
    Key in decimal point (.)
    Key in cents amount
    Press addition (+) key
Until all prices have been entered
Write down total price
Turn off calculator
```

What is flowchart?

- Pseudocode and flowchart are both popular ways of **representing algorithms.**
- Flowchart are popular because they graphically represent the program logic by a series of **standard geometric symbols** and **connecting lines.**
- Flowchart are relatively easy to learn and are an intuitive method of representing the flow of control in an algorithm.

What is pseudocode?

- Pseudocode and flowchart are both popular ways of representing algorithms.
- There is **no standard** pseudocode at present.
- Like many version of pseudocode, this version has certain conventions:
    1. Statements are written in simple English
    2. Each instruction is written on a separate line
    3. Keywords and indentation are used to signify particular control structures
    4. Each set of instructions is written from top to bottom, with only one entry and one exit
    5. Groups of statements may be formed into modules, and that module given a name

The Three Basic Control Structures

1. **Sequence**

   The sequence control structure is the straightforward execution of one processing step after another.

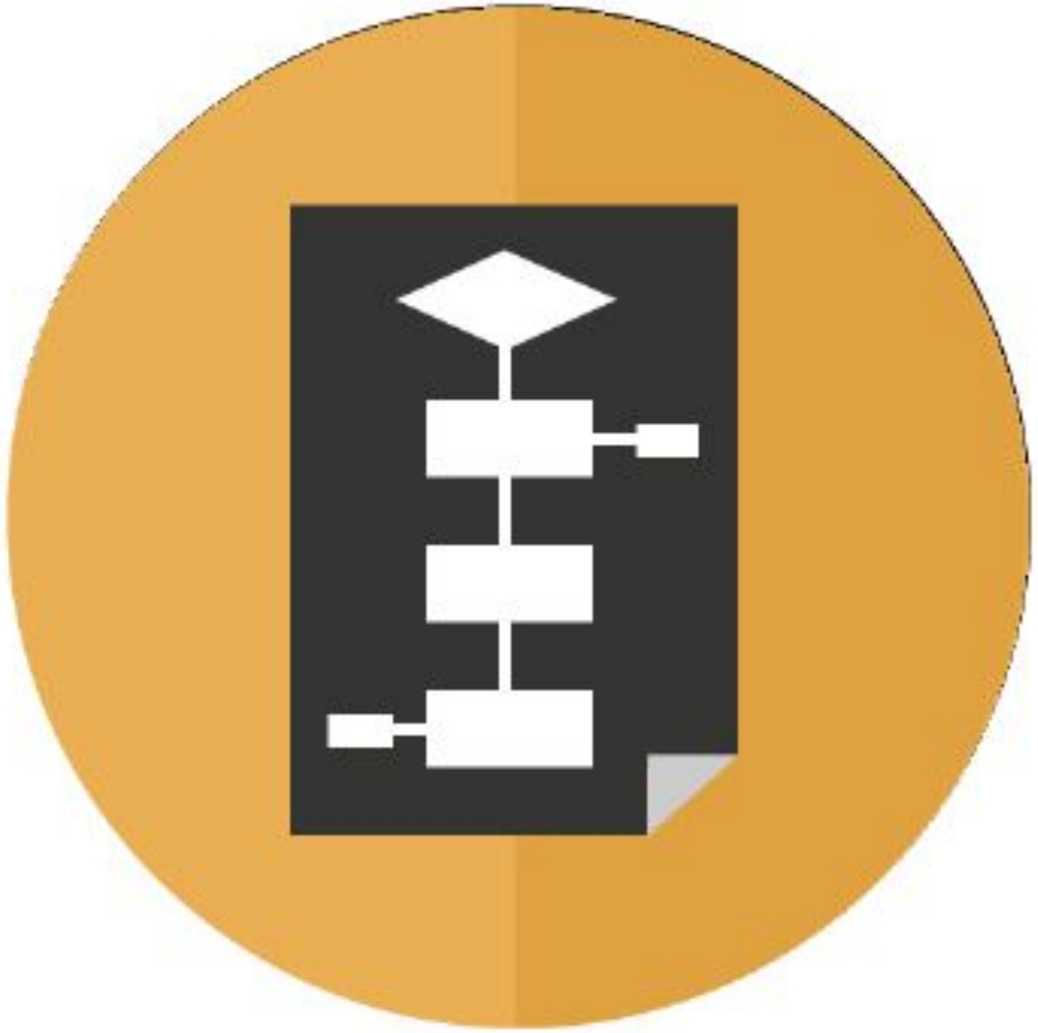
2. **Selection**

   The selection control structure is the presentation of a condition and the choice between two actions, the choice depending on whether the condition is **true** or **false**.

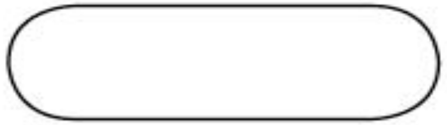
3. **Repetition**

   The repetition control structure can be defined as the presentation of a set of instructions to be performed repeatedly, as long as a condition is true.

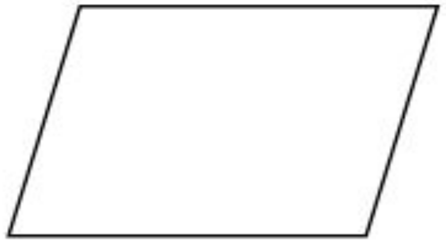
Introduction to Flowcharts



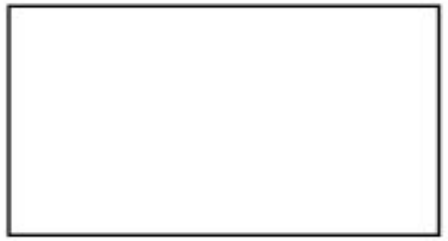
6 Standard flowchart symbols



Terminal symbol
The terminal symbol indicates the starting or stopping point in the logic. Every flowchart should begin and end with a terminal symbol.

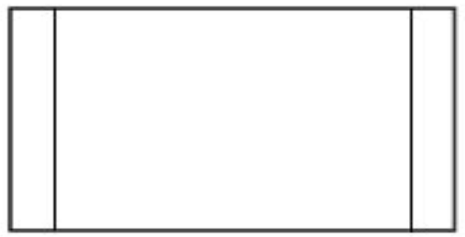


Input/Output symbol
The input/output symbol represents an input or output process in an algorithm, such as reading input or writing output.

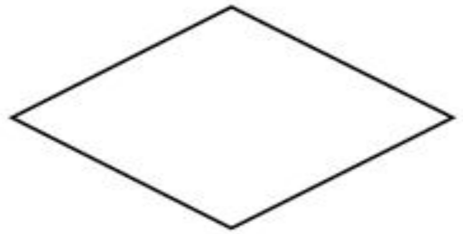


Process symbol
The process symbol represents any single process in an algorithm, such as assigning a value or performing a calculation. The flow of control is sequential.

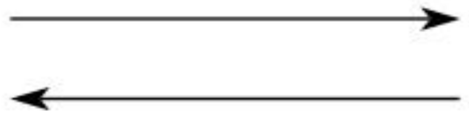
6 Standard flowchart symbols



Predefined process symbol
The predefined process symbol represents a module in an algorithm – that is, a predefined process that has its own flowchart.

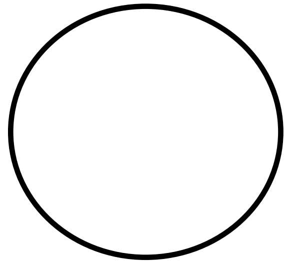


Decision symbol
The decision symbol represents a decision in the logic involving the comparison of two values. Alternative paths are followed, depending on whether the decision symbol is true or false.

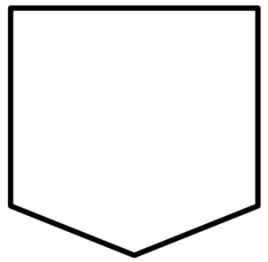


Flowlines
Flowlines connect various symbols in a flowchart, and contain an arrowhead only when the flow of control is not from top to bottom or left to right.

6 Standard flowchart symbols



Connector symbol

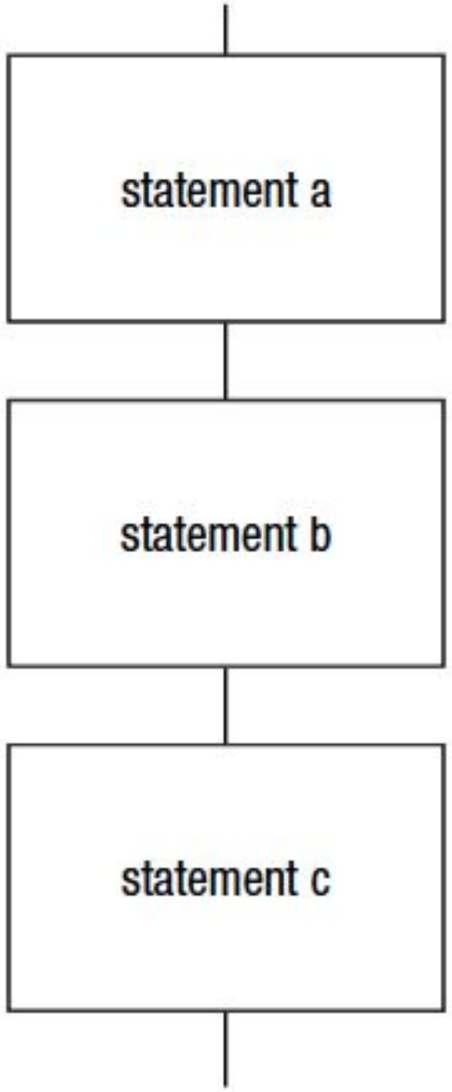


Off page connector symbol

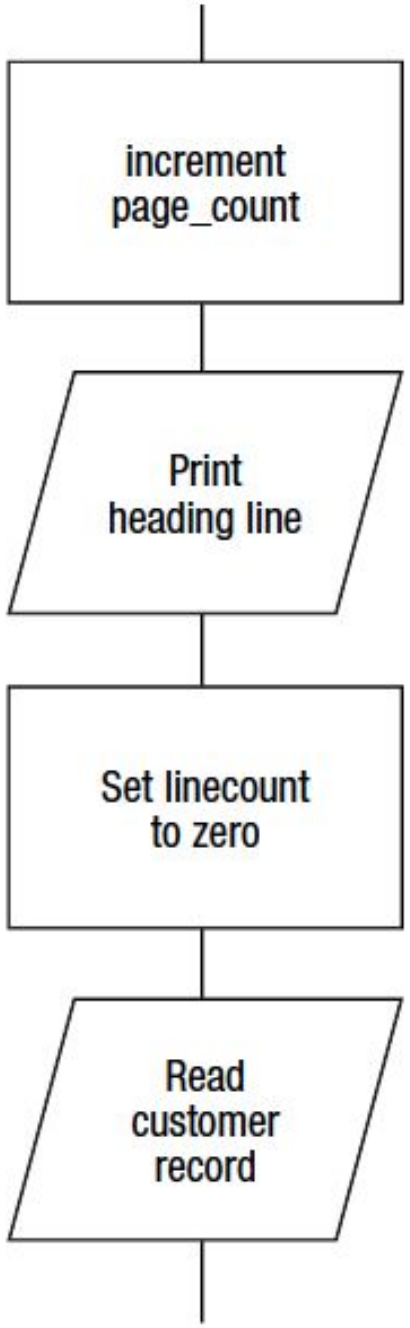
- In flowcharts, this symbol is typically small and is used as a **connector** to **show a jump** from one point in the process flow to another.
- Connectors are usually labeled with capital letters (A, B, AA) to show matching jump points

Flowchart : Sequence Control Structure

- A flowchart represents this control structure as a series of process symbols, one beneath the other, with one **entrance** and one **exit**



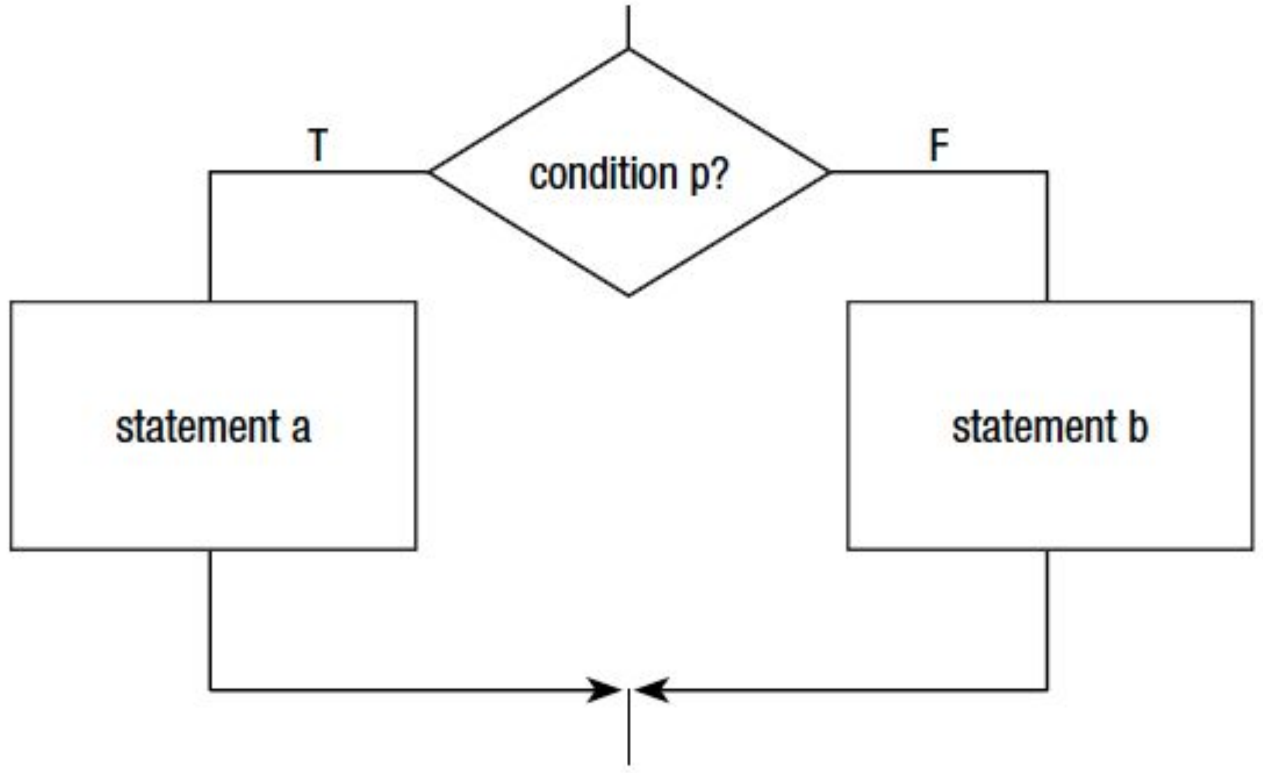
Sequence Flowchart Exampl



Flowcharts and The Selection Control Structure

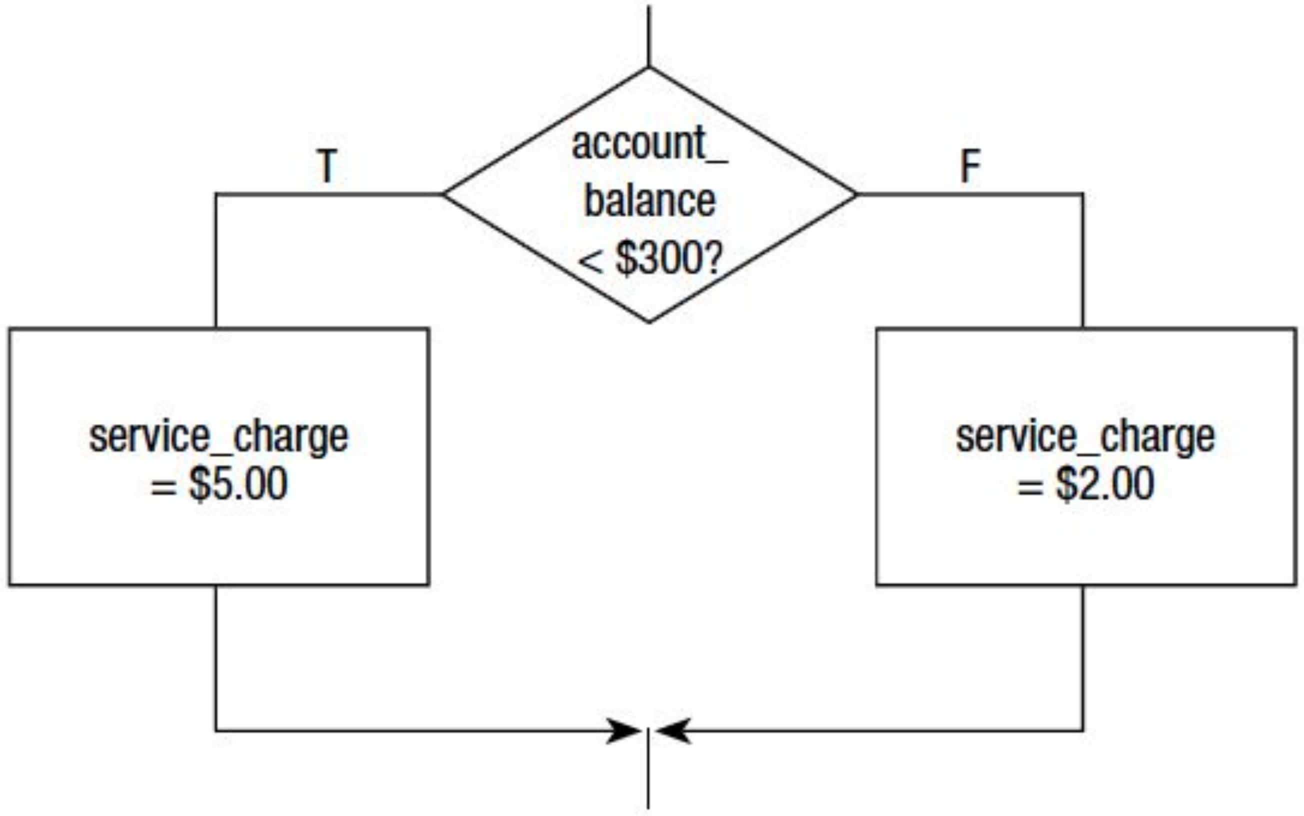
Selection Control Structure

- The selection control structure can be defined as the presentation of a condition, and the choice between two actions depending on whether the condition is **true** or **false**.



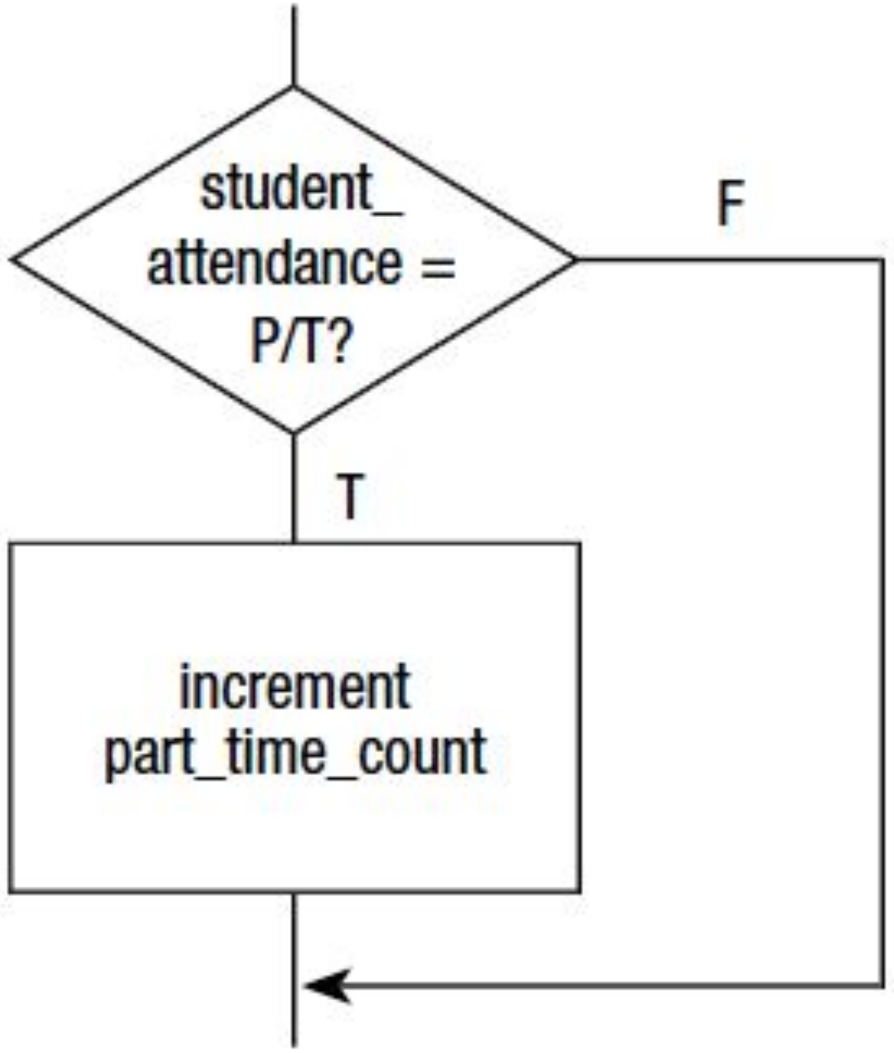
Simple IF Statement

- Simple selection occurs when a choice is made between two alternative paths, depending on the result of a condition being true or false.
- This structure is represented in a flowchart as follows:



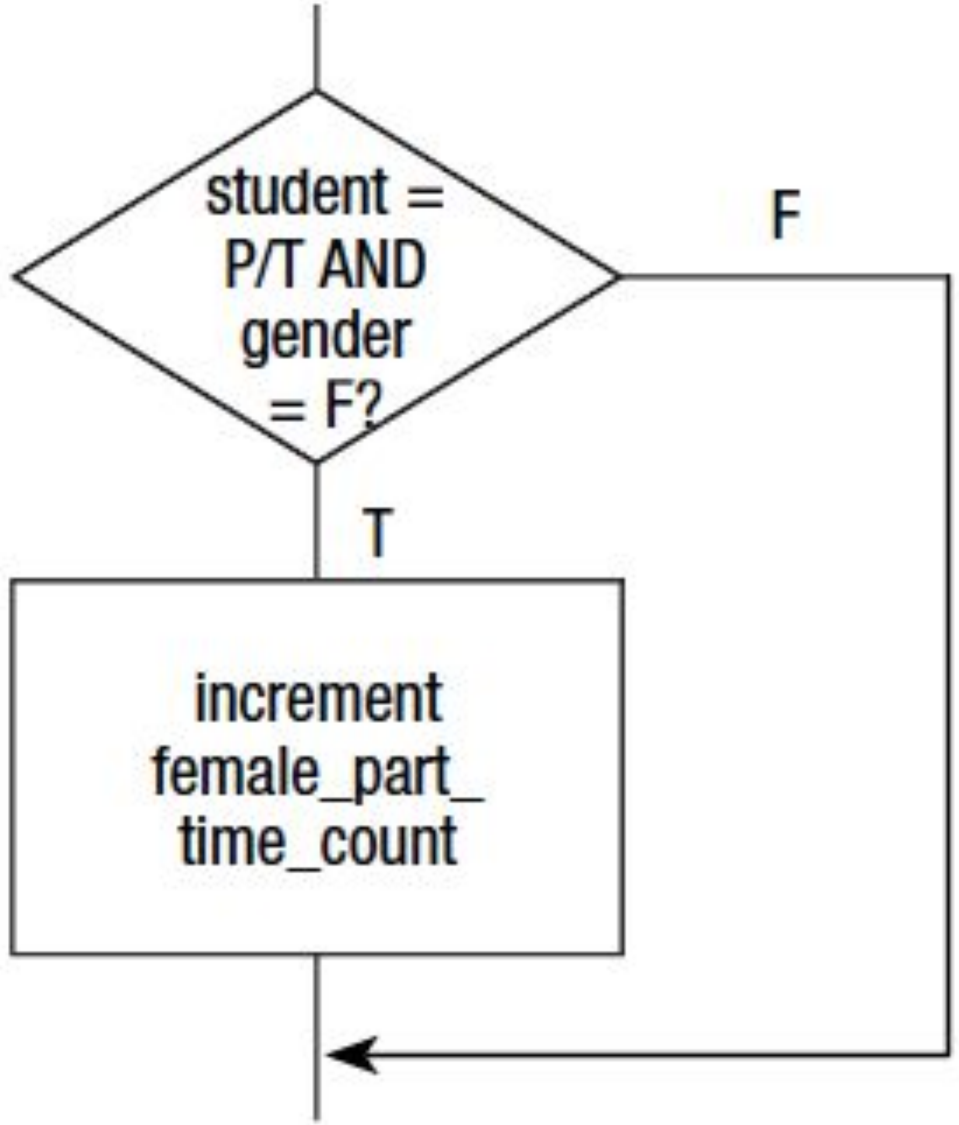
Null ELSE Statement

- The null ELSE structure is a **variation** of the simple IF structure.
- It is used when a task is performed only when a particular condition is true.
- If the condition is **false**, **no processing** will take place, and the IF statement will be bypassed.



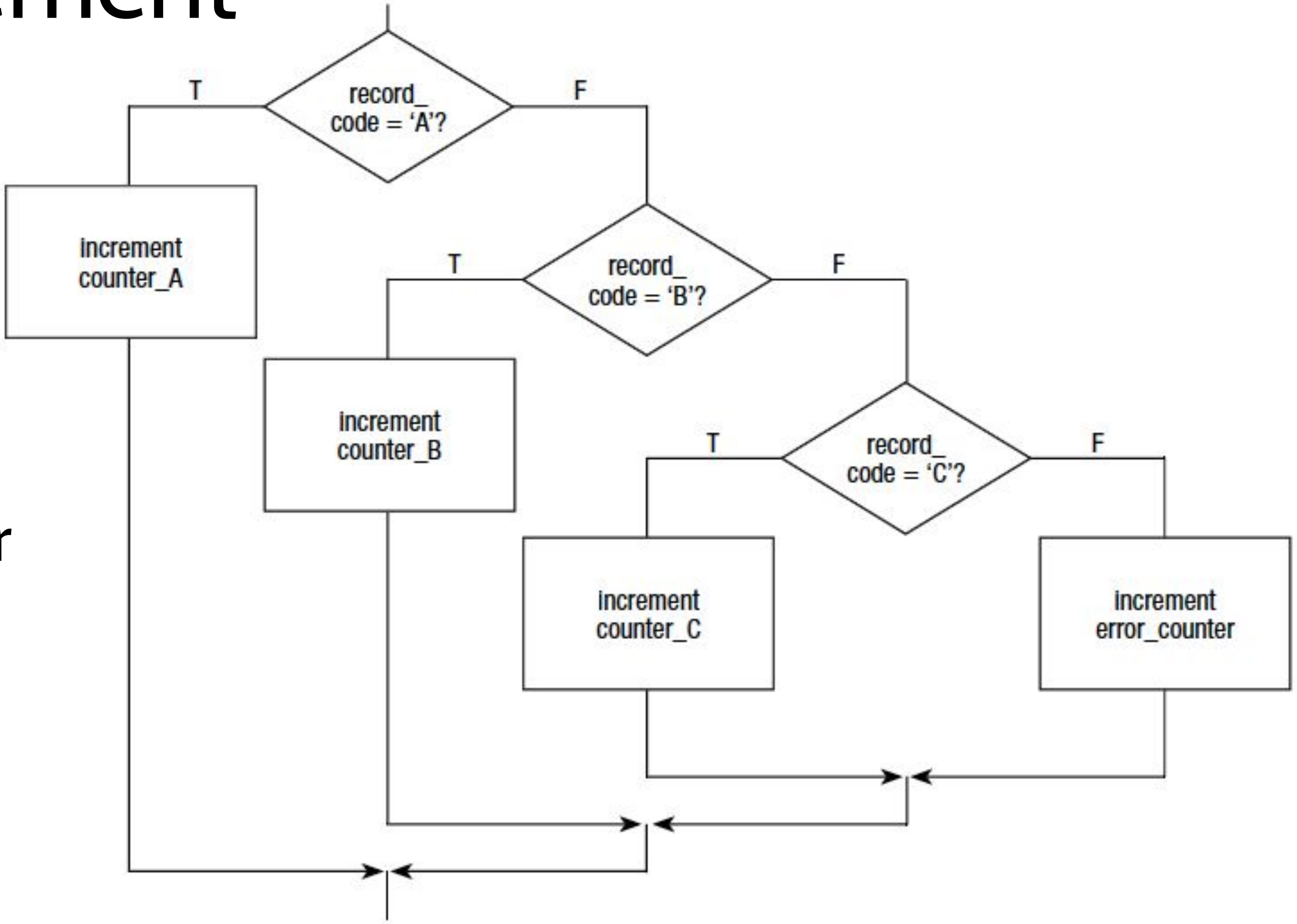
Combined IF Statement

- A combined IF statement is one that contains multiple conditions in the decision symbol, each connected with the logical operators **AND** or **OR.**
- If the connector **AND** is used to combine the conditions, then both conditions must be true for the combined condition to be true.


student = P/T AND gender = F?
F
T
increment female_part_time_count

Nested IF Statement

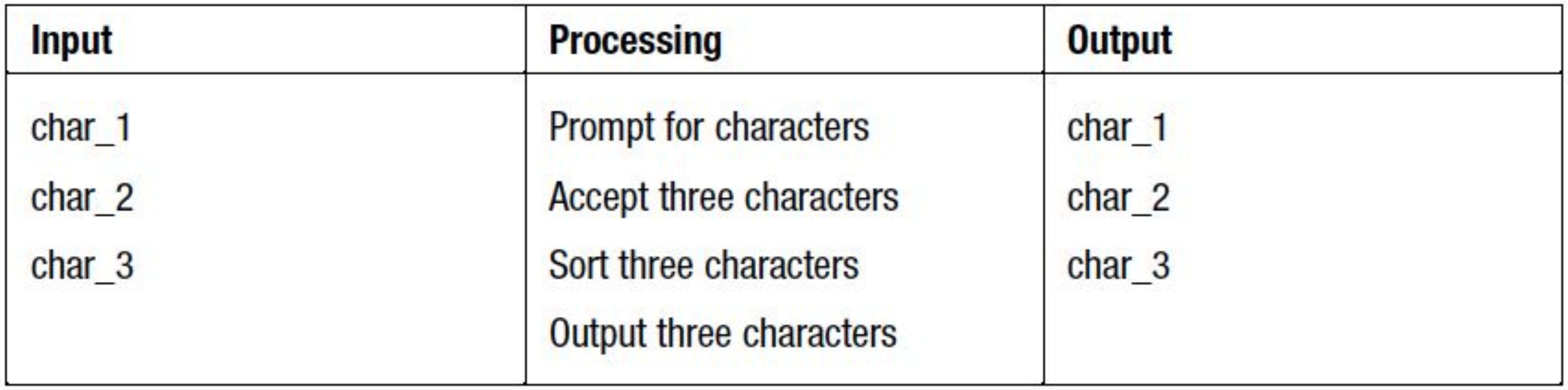
- The nested IF statement is used when a field is being tested for various values, with different action to be taken for each value.
- In a flowchart, this is represented by a **series** of decision symbols



Example 1: Read three characters

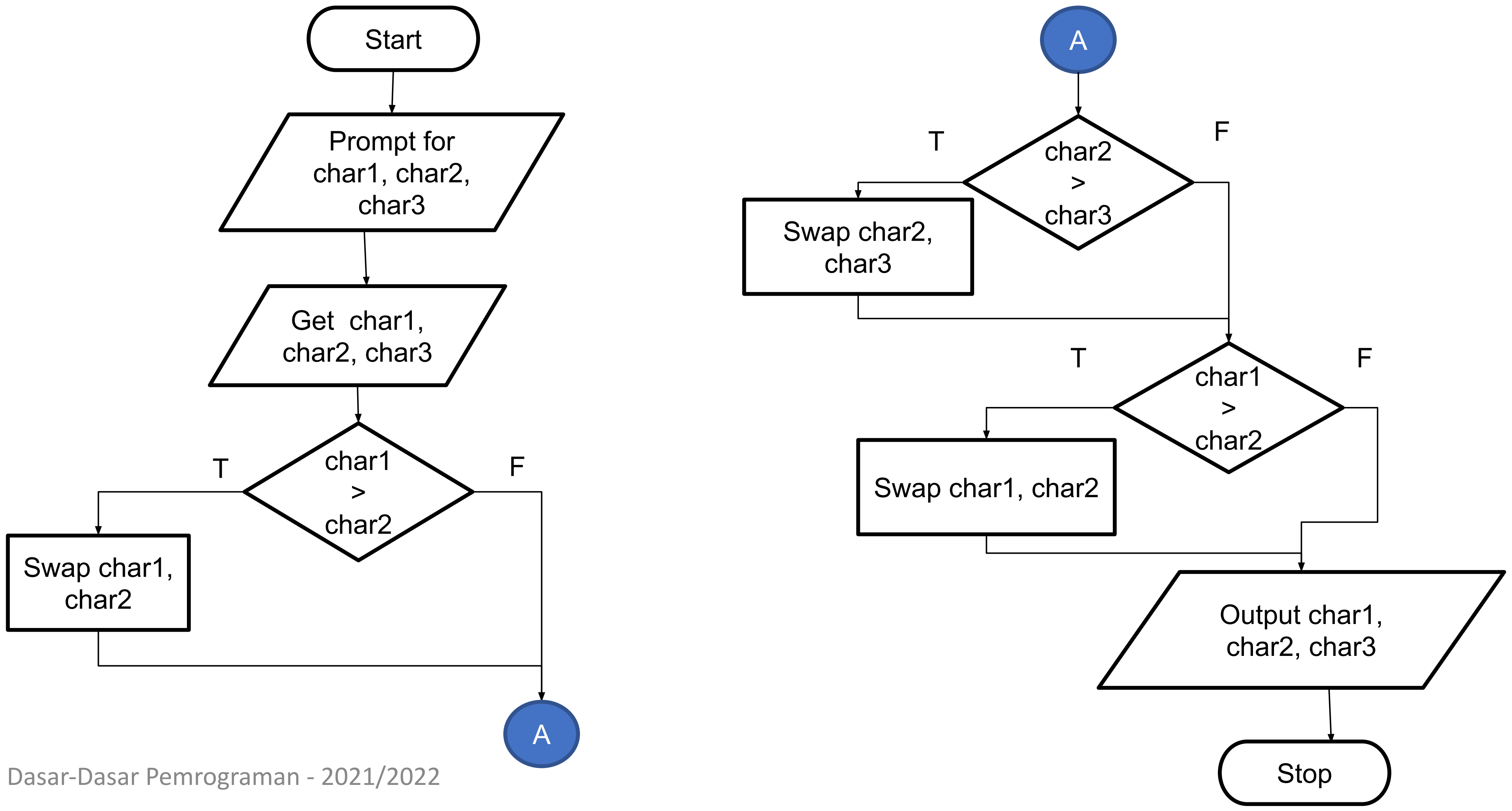
Design an algorithm that will prompt a terminal operator for three characters, accept those characters as input, sort them into **ascending** and output them to the screen.

Example 1: Read three characters



Input	Processing	Output
char_1	Prompt for characters	char_1
char_2	Accept three characters	char_2
char_3	Sort three characters	char_3
	Output three characters	

Example 1: Read three characters



Example 2: Process customer record

A program is required to <u>read</u> a customer's name, a purchase amount and a tax code. The tax code has been validated and will be one of the following:

0 – tax exempt (0 %)

1 – state sales tax only (3 %)

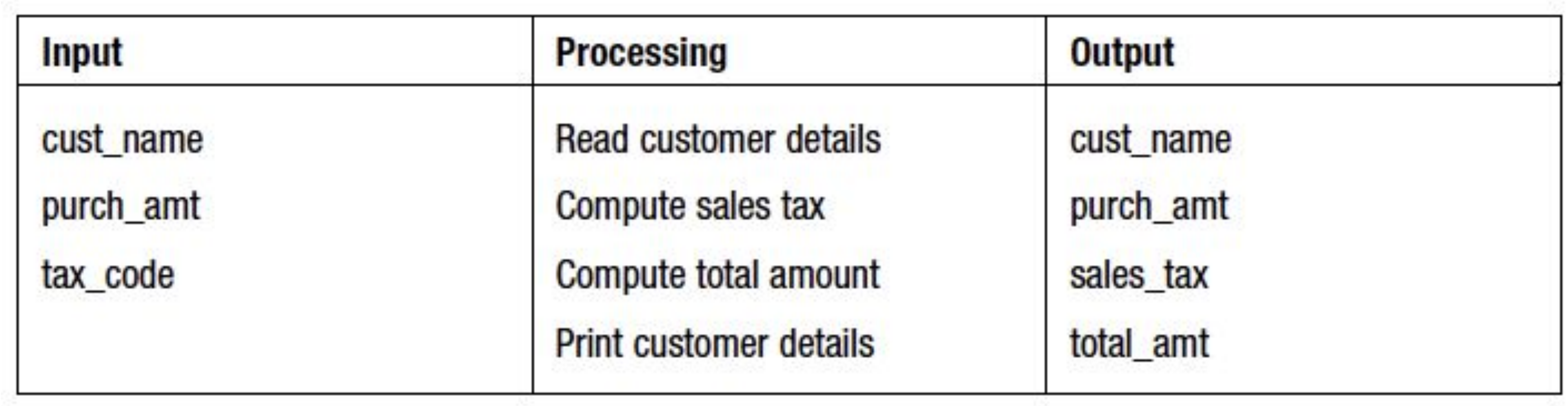
2 – federal and state sales tax (5 %)

3 – special sales tax (7 %)

The program must then <u>compute</u> the sales tax and the total amount due, and <u>print</u> the customer's name, purchase amount, sales tax and total amount due.

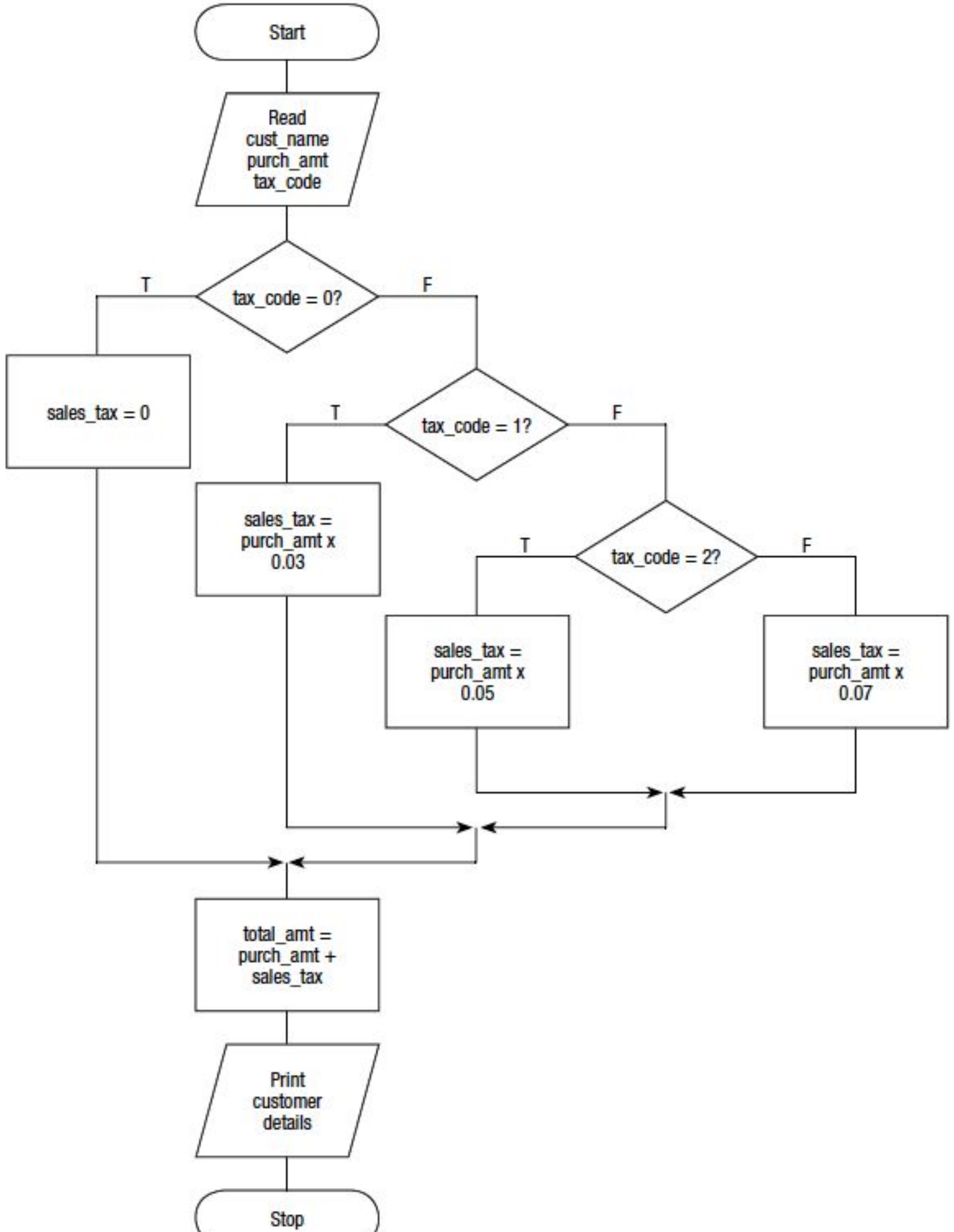
Example 2: Process customer record

Defining diagram



Input	Processing	Output
cust_name	Read customer details	cust_name
purch_amt	Compute sales tax	purch_amt
tax_code	Compute total amount	sales_tax
	Print customer details	total_amt

Example 2: Process customer record


Start
Read cust_name purch_amt tax_code
tax_code = 0?
T
F
sales_tax = 0
tax_code = 1?
T
F
sales_tax = purch_amt x 0.03
tax_code = 2?
T
F
sales_tax = purch_amt x 0.05
sales_tax = purch_amt x 0.07
total_amt = purch_amt + sales_tax
Print customer details
Stop

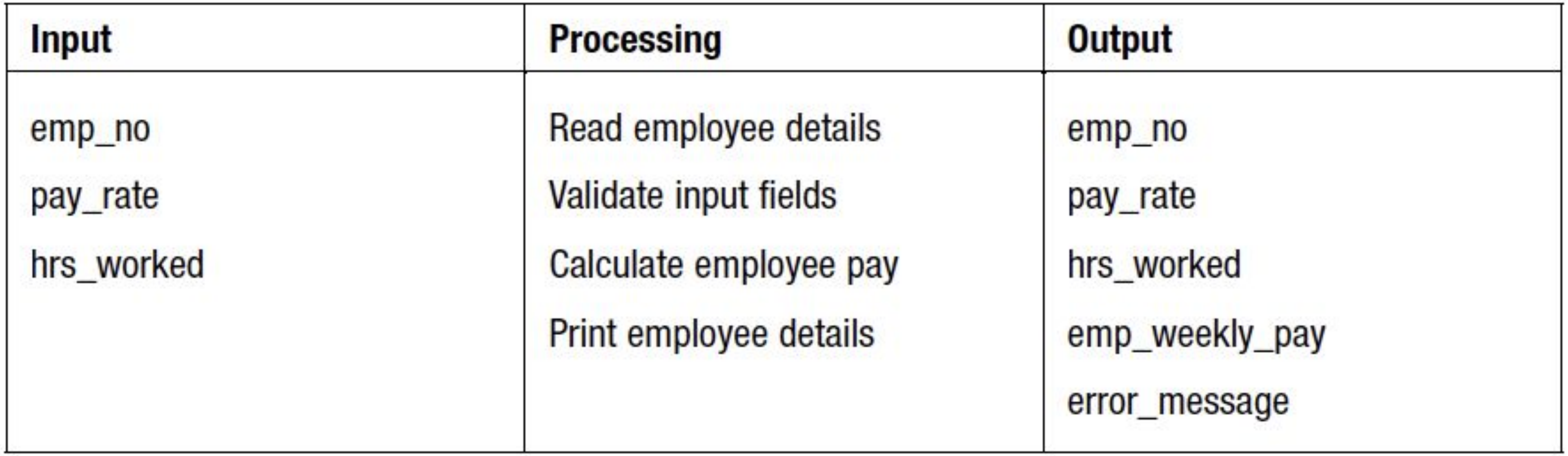
Example 3: Calculate employee's pay

A program is required by a company to <u>read</u> an employee's number, pay rate and the number of hours worked in a week. The program is then to <u>validate</u> the pay rate field and the hours work field and, if valid, <u>compute</u> the employee's weekly pay and then <u>print</u> it and the input data.

<u>Validation</u>: According to the company's rules, the maximum hours an employee can work per week is 60 hours, and the maximum hourly rate is $25.00 per hour. If the hours worked field or the hourly rate field is out of range, the input data and an appropriate message are to be <u>printed</u> and the employee's weekly pay is not to be calculated.

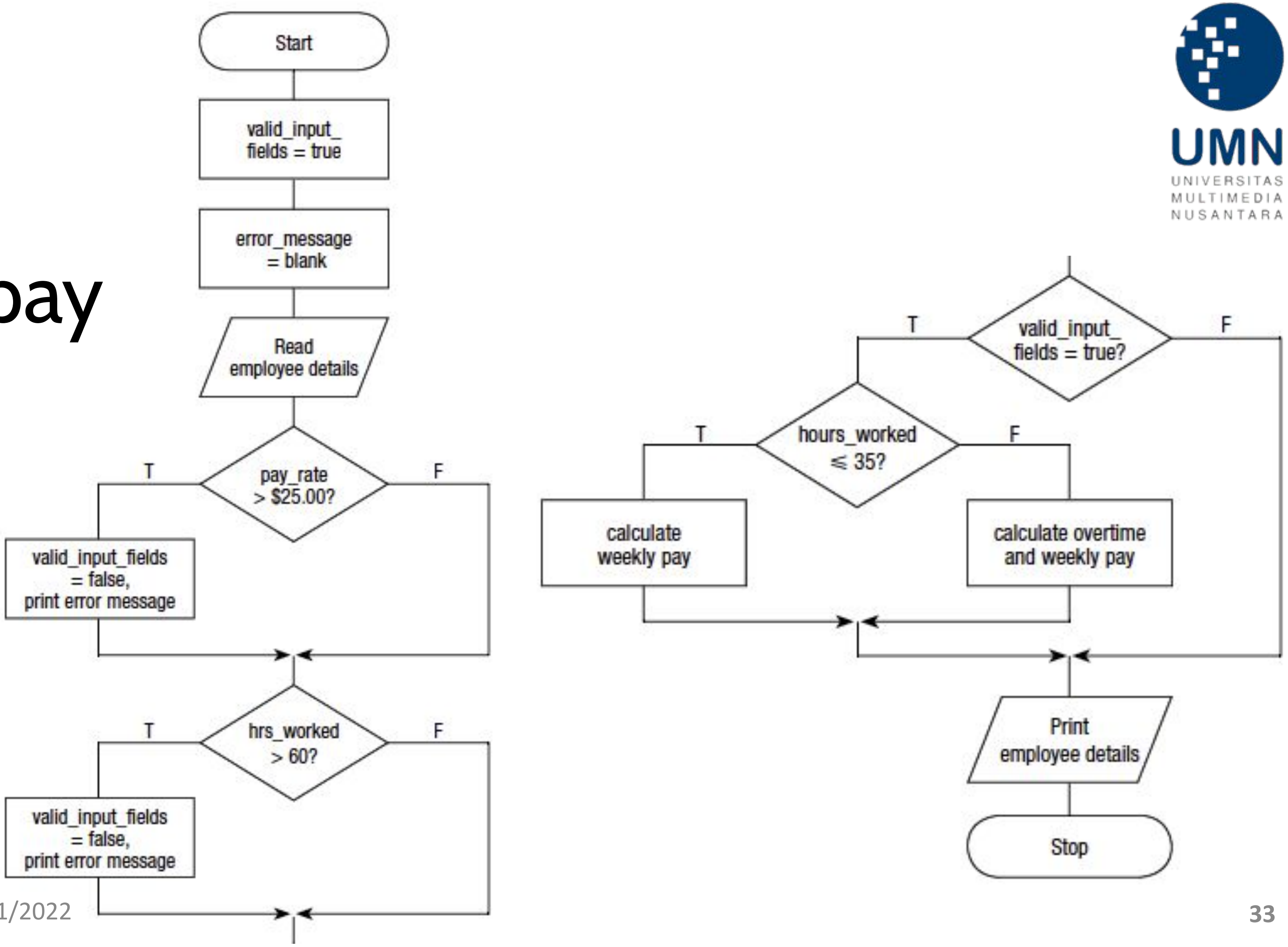
<u>Weekly pay calculation</u>: Weekly pay is calculated as hours worked times pay rate. If more than 35 hours are worked, payment for the overtime hours worked is calculated at time-and-a-half.

Example 3: Calculate employee's pay



Input	Processing	Output
emp_no	Read employee details	emp_no
pay_rate	Validate input fields	pay_rate
hrs_worked	Calculate employee pay	hrs_worked
	Print employee details	emp_weekly_pay
		error_message

Example 3: Calculate employee's pay

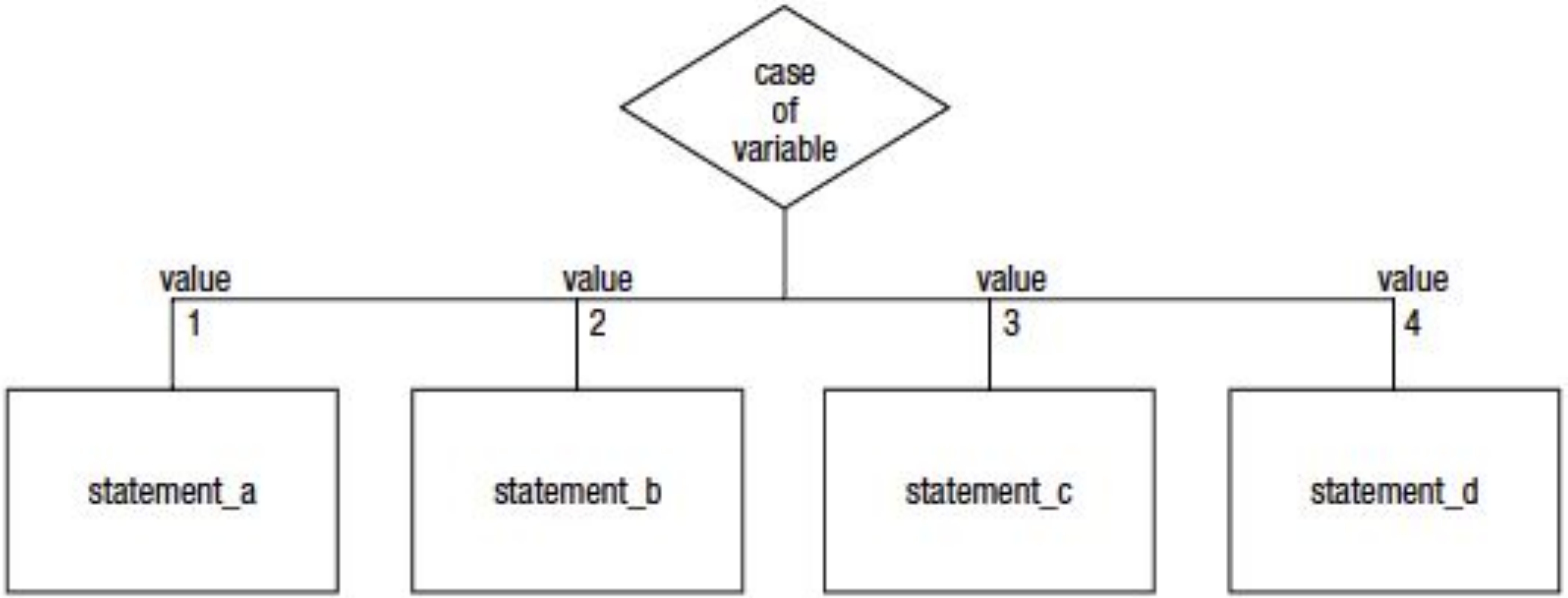


Start
valid_input_ fields = true
error_message = blank
Read employee details
pay_rate > $25.00?
T
F
valid_input_fields = false, print error message
hrs_worked > 60?
T
F
valid_input_fields = false, print error message
valid_input_ fields = true?
T
F
hours_worked ≤ 35?
T
F
calculate weekly pay
calculate overtime and weekly pay
Print employee details
Stop

The Case Structure

- The case control structure is **another way** of expressing a nested IF statement.
- It is not really an additional control structure, but one that extends the basic selection control structure to be a choice between multiple values.
- It is expressed in a flowchart by a decision symbol with a number of paths leading from it, depending on the value of the variable.

The Case Structure



Example : Process customer record

A program is required to read a customer's name, a purchase amount and a tax code. The tax code has been validated and will be one of the following:

0 – tax exempt (0 %)

1 – state sales tax only (3 %)

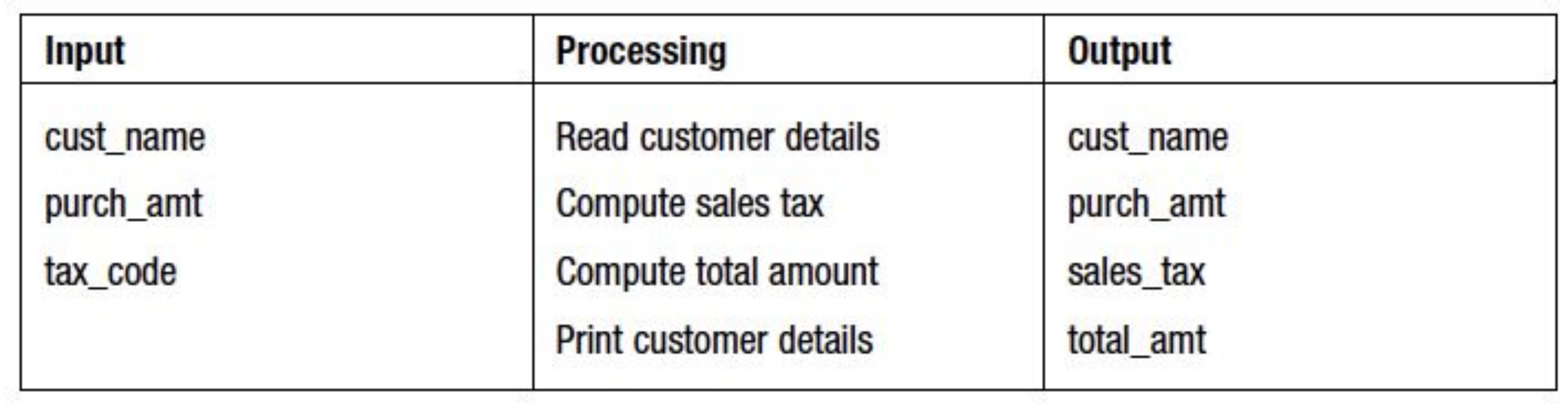
2 – federal and state sales tax (5 %)

3 – special sales tax (7 %)

The program must then compute the sales tax and the total amount due, and print the customer's name, purchase amount, sales tax and total amount due.

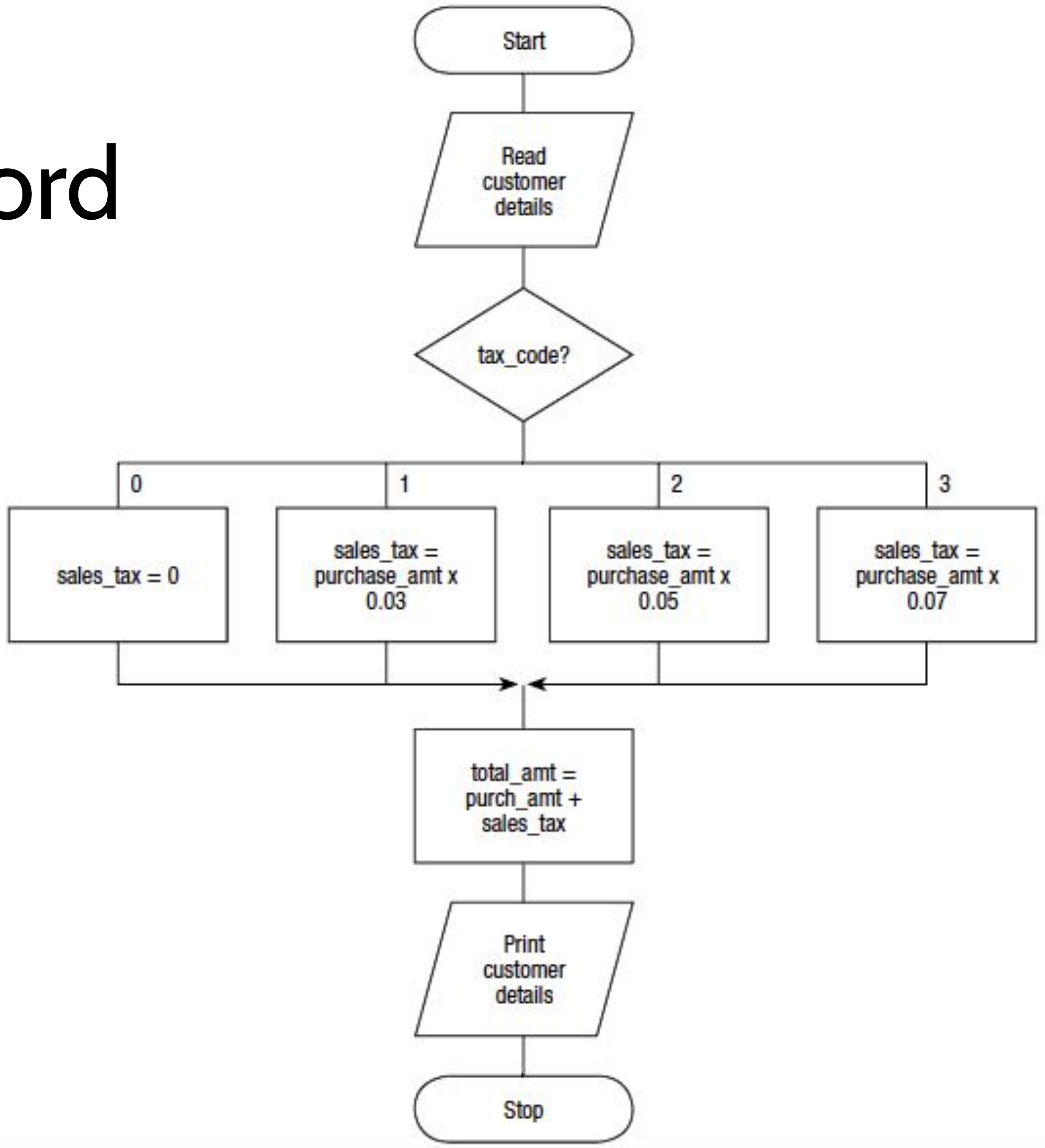
Example : Process customer record

Defining diagram



Input	Processing	Output
cust_name	Read customer details	cust_name
purch_amt	Compute sales tax	purch_amt
tax_code	Compute total amount	sales_tax
	Print customer details	total_amt

Example: Process customer record (In Case Structure Flowchart Diagram)


Start
Read customer details
tax_code?
0
1
2
3
sales_tax = 0
sales_tax = purchase_amt x 0.03
sales_tax = purchase_amt x 0.05
sales_tax = purchase_amt x 0.07
total_amt = purch_amt + sales_tax
Print customer details
Stop

Practice 1

Design an algorithm **in flowchart** that will <u>receive</u> two integer items from a terminal operator, and <u>display</u> to the screen their sum, difference, product, and quotient. Note that the quotient calculation (first integer divided by second integer) is only to be performed if the second integer does not equal zero.

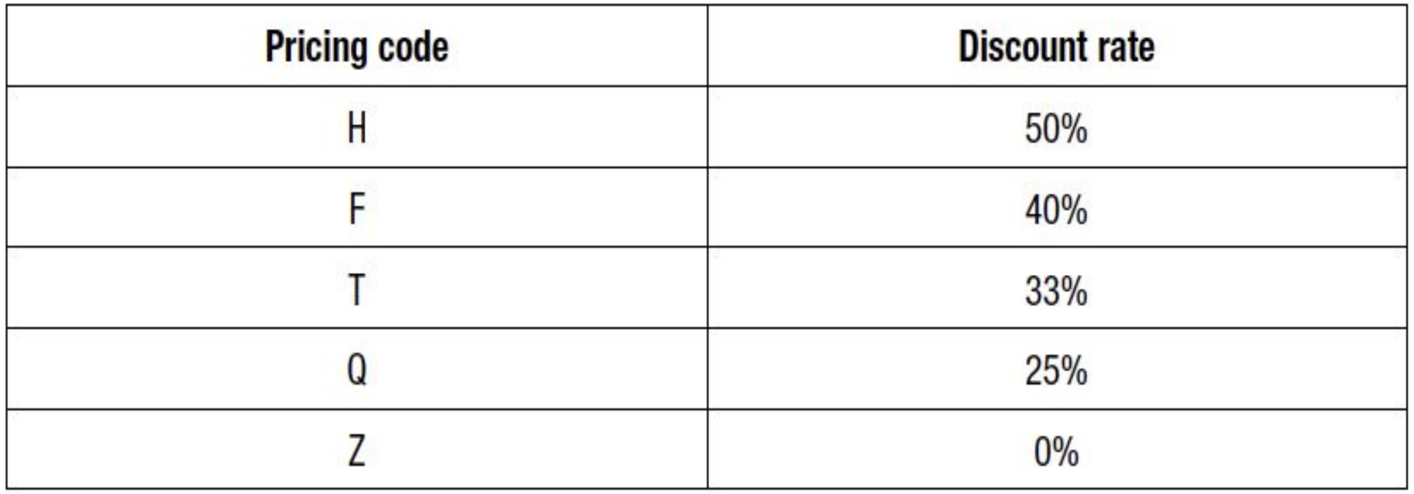
Practice 2

Design an algorithm **in flowchart** that will prompt an operator for a student's serial number and the student's exam score out of 100. Your program is then to match the exam score to a letter grade and print the grade to the screen. Calculate the letter grade as follows:

Exam score	Assigned grade
90 and above	A
80–89	B
70–79	C
60–69	D
below 60	F

Practice 3

Design an algorithm **in flowchart** that will prompt a terminal operator for the price of an article and a pricing code. Your program is then to calculate a discount rate according to the pricing code and print to the screen the original price of the article, the discount amount, and the new discounted price. Calculate the pricing code and accompanying discount amount as follows:



Pricing code	Discount rate
H	50%
F	40%
T	33%
Q	25%
Z	0%

If the pricing code is Z, the words ‘No discount’ are to be printed on the screen. If the pricing code is not H, F, T, Q, or Z, the words ‘Invalid pricing code’ are to be printed.

NEXT WEEK'S OUTLINE

1. Pseudocode with selection control structure
2. Desk checking
3. Exercises

REFERENCES

- Gaddis, Tony, 2019, Starting out with programming logic & design, Fifth edition, Pearson Education, Inc.

Visi

Menjadi Program Studi Strata Satu Informatika **unggulan** yang menghasilkan lulusan **berwawasan internasional** yang **kompeten** di bidang Ilmu Komputer (*Computer Science*), **berjiwa wirausaha** dan **berbudi pekerti luhur**.

INFORMATIKA
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Misi

1. Menyelenggarakan pembelajaran dengan teknologi dan kurikulum terbaik serta didukung tenaga pengajar profesional.
2. Melaksanakan kegiatan penelitian di bidang Informatika untuk memajukan ilmu dan teknologi Informatika.
3. Melaksanakan kegiatan pengabdian kepada masyarakat berbasis ilmu dan teknologi Informatika dalam rangka mengamalkan ilmu dan teknologi Informatika.